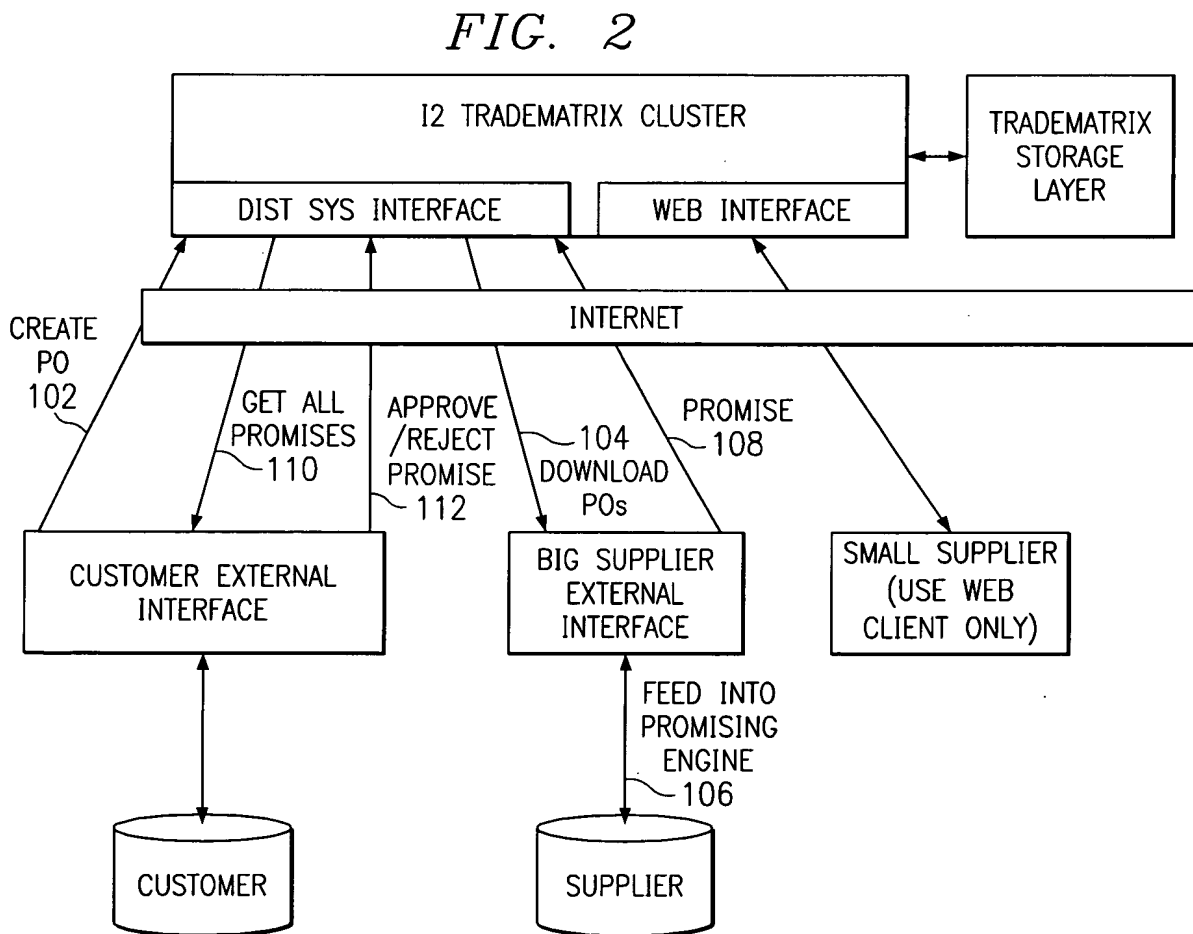
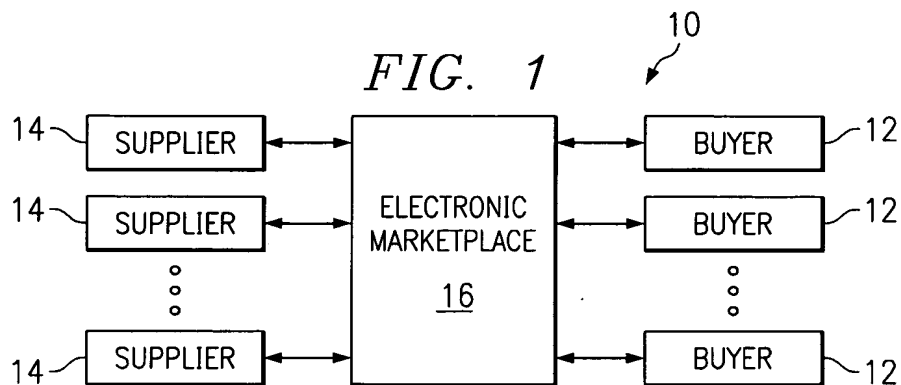
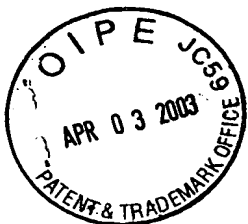


1/2



2/2

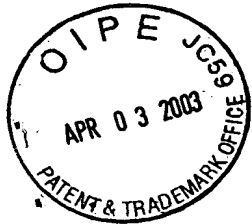


FIG. 3

```
public interface StateAggregator
{
    //Returns the resulting state based on composite states
    public State getAggregatedState (State [] compositeStates,
                                     GSContext gsc)
        throws StateAggregatorException;
    public String getAggregatorType()
        throws StateAggregatorException;
}

//Pseudo code illustrating the use of state aggregation
modifyPurchaseOrder (PO, ...)
{
    State [] lineItemStateArray;
    for every line Item
        State [] StateArray;
        for every schedule in line item
            process schedule to modify and pass through schedule
state transition
            StateArray.add (new state of this schedule)
        Endfor
        State ResultingLineItemState =

    POLineItemAggregator.getAggregatedState (StateArray)
        ModifyStateOfLineItem(lineItem, resultingLineItemState);
        LineItemStateArray.add(resultingLineItemState);
    Endfor
    State resultingPOState =

    POStateAggregator.getAggregatedState(lineItemStateArray);
        ModifyStateOfPO (PO, resultingPOState);
}
```